BVM-Tutorial 2010: BlueBerry
A modular, cross-platform, C++ application framework

Daniel Maleike, Michael Müller, Alexander Seitel, Marco Nolden, Sascha Zelzer
Overview

• General introduction
• Workbench overview
• Plug-in system and extension points
• Advanced topics
• Summary
Introduction

Main use-cases:

- Write your own plug-in for a MITK application
- Write your own application which can be extended by plug-ins

Features provided by BlueBerry:

- A plug-in system based on OSGi (as for example in Eclipse)
- Loose coupling of plug-ins via „Extension-Points“ (lazy-loading)
- A service registry together with some standard services (Preferences, Extension Registry, etc.)
- A highly customizable (GUI)-application framework
The Workbench - Overview

Menu contributions

Editors

Views
Workbench features:

- You can add arbitrary views and editors to your (or others) application
- Define *perspectives*, a layout of views and editors designed for specific tasks
- Add preferences with a GUI
- Use a generic selection service to communicate indirectly with other plug-ins
- Listen to workbench events (view activated, perspective changed, etc.)
- Customize the application layout by writing your own Workbench advisor classes
A plug-in must provide meta information about itself:

META-INF/MANIFEST.MF

Manifest-Version: 1.0
Bundle-Name: BlueBerry User Interface Plugin
Bundle-SymbolicName: org.blueberry.ui
Bundle-Version: 1.0.0
Bundle-Vendor: DKFZ, Medical and Biological Informatics
Require-Bundle: org.blueberry.osgi, ...
Bundle-Activator: berry::WorkbenchPlugin

- See the public attributes in IBundleManifest for a list of available keys.
Extension points can be used to lazily provide or collect information:

**plugin.xml**

```xml
<extension point="org.blueberry.ui.views">
  <category id="org.mitk.views.general" name="MITK General"/>
  <view id="org.mitk.views.datamanager" name="Datamanager" category="org.mitk.views.general" icon="resources/datamanager.xpm" class="QmitkDataManagerView" />
</extension>
```

- See the [Extension-Point Reference](#) for a list of available extensions
The Plug-In System

- Meta information
- Internal code
- Exported code
- Extension points

- See the PluginStyleGuide for a more in-depth explanation
Advanced Topics

**Perspectives:**
- Define an optimal layout for logically connected views
- Read more at [Using Perspectives](#)

**Preferences:**
- Persist user and internal preferences
- Read more at [The Preferences Service](#)

**Selection Service:**
- Publish or subscribe to generic selection events
- Read more at [Using the Selection Service](#)

**Custom Applications:**
- Provide custom welcome screens, an initial perspective, etc.
- Read more at [WorkbenchAdvisor](#)
Summary

Plugins

- Can be lazily loaded through extension points
- Can extend the Platform‘s capabilities
- Can extend the capabilities of other plug-ins

The Workbench

- Can be customized by changing the set of plug-ins and defining perspectives
- Provides useful standard services (Preferences, Selections, ...)

Further information

- [http://www.mitk.org/wiki/BlueBerry](http://www.mitk.org/wiki/BlueBerry)
Thank you!

Any questions?